Applicant Joseph G. Barrett
Title. Securing An Access Provider.

2152
#3

Docket No.: 06975-074001

09/666,104

SECURING AN ACCESS PROVIDER

TECHNICAL FIELD

This invention relates to securing an access provider. More particularly, this invention relates to detecting and preventing denial of service attacks on an access provider.

BACKGROUND

Access providers have proven susceptible to various attacks by computer hackers. In a type of computer attack known as a denial of service attack, a hacker attempts to deny service to legitimate users of online computer services. For instance, a hacker may send a high number of illegitimate access requests to an accessible computer system of an access provider (hereinafter access provider), causing the access provider to dedicate its resources to handling the illegitimate access requests rather than handling legitimate access requests from legitimate users. In this manner, legitimate users may be denied access to an online computer service enabled by the access provider because of the influx of illegitimate access requests sent by the hacker. This type of attack is commonly known as a synchronize (SYN) flood.

Another type of a computer attack occurs when a hacker attempts to gain unauthorized access to an online computer service through an access provider. In this type of attack, the hacker uses a client to attempt to establish an unauthorized connection with the access provider. For instance, the hacker begins by identifying a logon identification known to be valid. The hacker then attempts to crack the password associated with the valid logon identification. For instance, the hacker may use a computer program to associate several passwords with the logon identification in rapid succession, repeatedly attempting to establish a connection with the access provider using the known logon identification and one of the associated passwords. This type of attack may tax processing resources to effectively deny legitimate users access to the online computer service.

When subject to such attacks, access providers may be forced to cease operation.

SUMMARY

In one general aspect, securing an access provider includes monitoring communications with at least one access provider for a partially-completed connection

transaction and terminating the partially-completed connection transaction when the partially-completed connection transaction remains in existence for a period of time that exceeds a threshold period of time.

Embodiments may include one or more of the following features. For example, the monitoring may include detecting partially-completed connection transactions initiated by an access requestor, measuring the period of time that a partially-completed connection transaction remains in existence, and comparing the period of time with the threshold period of time.

The monitoring also may include monitoring communications with at least one access provider based on TCP communications for partially-completed connection transactions. The monitoring may include monitoring a process whereby an access requestor sends a SYN request and the access provider sends a SYN acknowledgement. The monitoring may include monitoring communications with a plurality of access providers for partially-completed connection transactions.

The monitoring also may include detecting partially-completed connection transactions that occur when an access requestor initiates a connection transaction and the access requestor subsequently fails to send a reply. More particularly, the monitoring may include detecting partially-completed connection transactions that occur when an access requestor initiates a connection transaction based on a return address that differs from an actual return address of the access requestor. This particular instance may occur when the return address is an Internet protocol address that differs from the actual return address of the access requestor.

Where the access requestor is a client and the access provider is a host, the monitoring may include detecting partially-completed connection transactions between at least one client and at least one host, and/or detecting partially-completed connection transactions between at least once client and a plurality of hosts and/or detecting partially-completed connection transactions between a plurality of clients and at least one host.

The terminating may include resetting a communication port located on the access provider. When the threshold period of time is configurable, the terminating also may include terminating the partially-completed connection transaction when the partially-completed connection transaction remains in existence for a period of time that exceeds a configurable threshold period of time.

These general and specific aspects may be implemented using a system or method or combination of system and method.

The details of one or more embodiments are set forth in the accompanying drawings and the description below. Other features and advantages will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

Fig. 1 is a block diagram that illustrates the physical level of a system for securing an access provider.

Fig. 2 is a block diagram that illustrates the logical level of a system for securing an access provider of Fig. 1.

Fig. 3 is a block diagram that illustrates components included in a switch such as those shown by Figs. 1 and 2.

Fig. 4 is a block diagram that illustrates components included in a monitoring component of the switch of Fig. 3.

Fig. 5 is a flowchart of a process for securing an access provider, which may be performed by the systems shown by Figs. 1-4.

Fig. 6 is a flowchart of a process for monitoring the access provider for partially-completed connection transactions as part of the process of Fig. 5.

Like reference symbols in the various drawings may indicate like elements.

DETAILED DESCRIPTION

Fig. 1 is a block diagram that illustrates the physical level of an accessible computer system 100. Fig. 1 shows multiple access requestors 110, the Internet 130, multiple routers 150, switch 170, multiple access providers 190, and multiple communication links 120, 140, 160, and 180.

An access requestor 110 may include a client, and may be embodied in a general-purpose computer (e.g., a personal computer), a special-purpose computer, a workstation, a server, a personal digital assistant, an electronic organizer, a mobile phone, a pager, a device, a component, or other physical or virtual equipment or some combination thereof, any of which may be programmed or configured to respond to and execute instructions in a defined

manner. Access requestors 110 are connected to the Internet 130 by communication links 120.

The Internet 130 is an example of a delivery network that may be used to enable communications to/from access requestors 110. Other examples of a delivery network may include the World Wide Web, wide area networks (WANs), local area networks (LANs), analog or digital wired and wireless telephone networks (e.g. Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN), and Digital Subscriber Lines (xDSL)), radio, television, cable, satellite, and/or any other delivery mechanism for carrying data. The Internet 130 is generally connected to one or more routers 150 by communication links 140.

Each router 150 generally includes a computer processor, computer software, a hardware device, other physical or virtual equipment or some combination of these elements that is capable of receiving, processing and transmitting information. In general, each router 150 routes communications between one or more access requestors 110 and one or more access providers 190. Communications received from an access provider 190 are generally routed to an access requestor 110 through the Internet 130. Communications received from an access requestor 110 are generally routed to an access provider 190 through a switch 170. More specifically, each router 150 receives a data packet and/or data request from access requestor 110 and routes the data packet and/or data request through switch 170 to one or more of the access providers 190 based on predefined criteria or algorithms. The routers 150 are connected to switch 170 by communication links 160.

Switch 170 generally includes one or more hardware components and one or more software components. It is capable of receiving a unit of data and of transmitting the received data to one or more access providers 190 or routers 150 based on predefined criteria or algorithms. Switch 170 may perform load balancing algorithms such as hashing techniques to avoid overwhelming any particular router 150 or access provider 190. Switch 170 also may perform the functions of the router 150 as a separate or integrated component or device. Additionally or alternatively, switch 170 may include one or more processors and one or more storage and memory devices, such as internal memory. The switch 170 is connected to multiple access providers 190 by communication links 180.

An access provider 190 may be any software or hardware capable of providing access by an access requestor 110 to desired information or services. For instance, an access

provider 190 may include a host, and it may be embodied in a general-purpose computer (e.g., a personal computer) or a special-purpose computer capable of communicating with one or more access requestors 110 by responding to and executing instructions in a defined manner. Other examples of an access provider 190 include a special-purpose computer, a work station, a server, a device, a component, other physical or virtual equipment or some combination of these elements that is capable of responding to and executing instructions as described.

Communication links 120, 140, 160 and 180 may include, for example, a wired, wireless, cable or satellite communication pathway.

Fig. 2 is a block diagram that illustrates a logical level of the system 100 illustrated in Fig. 1. Fig. 2 shows multiple access requestors 110, switch 170, and multiple access providers 190. In this figure, switch 170 may be representative of one or more of Internet 130, router 150 and switch 170, or some combination there between such as that described in Fig. 1.

An access requestor 110 is generally used to establish a physical or non-physical electronic connection with an access provider 190. Connections may be established on various levels using various protocols. For instance, a connection may be established on Level III (e.g., a packet based level), on Level IV (e.g., a protocol data unit based level with flow control and error correction) or on some other level using an appropriate protocol capable of establishing a connection between an access requestor 110 and an access provider 190. More specifically, examples of protocols include Transmission Control Protocol (TCP), Internet Protocol (IP), TCP/IP, User Datagram Protocol (UDP), and UDP/IP.

Access protocols are observed to establish a connection. In an exemplary Level IV protocol, an access requestor 110 sends an access request through switch 170. The request is routed to one of the access providers 190, which responds to the access request by sending an acknowledgement that is routed back to the access requestor 110 through switch 170. When the access requestor 110 receives the acknowledgement sent by the access provider 190, the access requestor 110 generates an acknowledgement that is sent back to the access provider 190 through switch 170. The completion of this transaction establishes a connection between the access requestor 110 and the access provider 190.

For purposes of this detailed description, the term connection transaction is used to describe one or more of sending, receiving, or exchanging the units of data necessary to use a

protocol (e.g., TCP, IP, UDP, TCP/IP, and UDP/IP) to establish a communication link (e.g., wired, wireless, cable, and satellite) between the access requestor 110 and access provider 190. One example of a connection transaction results in a TCP connection between the access requestor 110 and the access provider 190, where procedures to establish a connection transaction use the synchronize (SYN) control flag and involve an exchange of three messages. In this example, an access requestor 110 sends an access request (SYN REQ) to an access provider 190 through switch 170. The access provider 190 responds to the access requestor 110 through switch 170 with an acknowledgement (SYN ACK). Then, the access requestor 110 sends an acknowledgement (ACK) to access provider 190 via switch 170. Other connection transactions between access requestor 110 and access provider 190 through switch 170 are also possible and can result in different types of connections (e.g., IP, TCP/IP, UDP, and UDP/IP).

For purposes of this detailed description, the term partially-completed connection transaction is used to describe one or more of sending, receiving, or exchanging data that is necessary to establish a connection transaction under a protocol (e.g., TCP, IP, UDP, TCP/IP, and UDP/IP) but that is insufficient to establish a communications link (e.g., wired, wireless, cable, and satellite) between the access requestor 110 and access provider 190. One example of a partially-completed connection transaction occurs during an attempt to establish a TCP connection between access requestor 110 and access provider 190. In this example, an access requestor 110 sends an access request (SYN REQ) to an access provider 190 through switch 170. In response to the access request, the access provider 190 opens a communication port and sends an acknowledgement (SYN ACK) to the access requestor 110 through switch 170. The access requestor 110 fails to send a reply (ACK) to the access provider 190 via switch 170. This failure to send a reply results in the partially-completed connection transaction.

Another type of partially-completed connection transaction may occur when an illegitimate access requestor 110 initiates a connection transaction based on a spoofed return address that differs from an actual return address of the illegitimate access requestor 110. In this instance, the illegitimate access requestor 110 sends an access request (SYN REQ) to the access provider 190 via switch 170 using the spoofed return address. In response to the access request, the access provider 190 opens a communication port. Then, because the access request is spoofed, the access provider 190 sends an acknowledgement (SYN ACK) to

the spoofed return address, which differs from the actual return address of the illegitimate access requestor 110. Thereafter, no reply (ACK) is generated by the illegitimate access requestor 110, which does not even receive the acknowledgement (SYN ACK) sent by the access provider 190 to the spoofed return address. Thus, the attempted TCP connection is only a partially-completed connection transaction. In this example, the spoofed return address may be Internet protocol (IP) addresses which is capable of identifying each sender or receiver of information across the Internet 130.

When a partially-completed connection transaction occurs, the communication port that was opened on the access provider 190 remains opened, awaiting completion of the initiated connection transaction. Consequently, one of a finite number of communication ports on access provider 190 is used. Other partially-completed connection transactions between access requestor 110 and access provider 190 may occur on other levels (e.g., Level III) using other types of protocols (e.g., IP, TCP/IP, UDP, and UDP/IP).

Fig. 3 is a block diagram that illustrates logical components of switch 170. As shown, the switch 170 includes the components necessary to detect and prevent a hacker attack on access providers 190. In particular, switch 170 includes a monitoring component 310 and a terminating component 320, which generally include one or more elements embedded in software modules, but may be embodied in physical devices connected to one another or may be embodied in some combination of software modules and physical devices. In other implementations, the components illustrated in Fig. 3 may be resident on an access provider 190.

The monitoring component 310 is generally structured and arranged to monitor communications with at least one access provider 190 for partially-completed connection transactions. In one example, the monitoring component 310 may be structured and arranged to detect partially-completed connection transactions by monitoring communications performed to establish a connection involving an access provider 190 based on TCP protocol. In another example, the monitoring component 310 may be structured and arranged to monitor communications with several access providers 190 to detect partially-completed connection transactions. Additionally or alternatively, monitoring component 310 may be programmed to recognize partially-completed connection transactions based on other criteria, or other partially-completed connection transaction types altogether (e.g., IP, TCP/IP, UDP,

and UDP/IP). The monitoring component 310 may be preconfigured or it maybe programmable, as will be described in Fig. 4.

The terminating component 320 is generally structured and arranged to terminate partially-completed connection transactions when the partially-completed connection transactions remain in existence for a period of time that exceeds a threshold period of time.

Referring to Fig. 4, the monitoring component 310 may include a detection component 410, a measuring component 420, and a comparing component 430. The detection component 410 is generally structured and arranged to detect partially-completed connection transactions initiated by an access requestor 110. The detection component 410 is generally programmable and capable of recognizing when a partially-completed connection transaction occurs. For example, detection component 410 may be programmed to recognize a partially-completed connection transaction that occurs when an access requestor 110 initiates a connection transaction and the access requestor 110 subsequently fails to send a reply. In one scenario described above with respect to Fig. 2, the detection component 410 may detect partially-completed connection transactions that occur when an illegitimate access requestor 110 initiates a connection transaction based on a spoofed return address (e.g., IP address) that differs from an actual return address (e.g., IP address) of the illegitimate access requestor 110.

Where the access requestor 110 includes one or more clients and the access provider 190 includes one or more hosts, the detection component 410 is capable of detecting partially-completed connection transactions between at least one client and at least one host. Additionally or alternatively, where the access requestor 110 includes one or more clients and the access provider 190 includes one or more hosts, the detection component 410 may be capable of detecting partially-completed connection transactions between at least one client and multiple hosts and/or between multiple clients and at least one host. The detection component 410 generally communicates with to the measuring component 420.

The measuring component 420 is generally structured and arranged to measure the period of time that a partially-completed connection transaction remains in existence. The measuring component 420 is generally programmable. Measuring component 420 may be implemented using a processor and an internal memory for measuring and recording the period of time. It may be implemented using software performed by a processor, or it may be implemented using some combination of hardware and software. In the example of a

partially-completed connection transaction based on TCP communications, measuring component 420 measures a period of time that starts when an access request (SYN REQ) is first received. Alternatively, measuring component 420 may measure a period of time that starts when the access provider 190 opens a communication port, perhaps starting when the acknowledgement (SYN ACK) is sent in response to an access request (SYN REQ). In this way, measuring component 420 may measure and record the period of time that the communication port remains in existence.

Measuring component 420 is capable of measuring the period of time across multiple access providers 190. The measuring component 420 communicates with the comparing component 430.

The comparing component 430 is generally structured and arranged to compare the period of time measured by the measuring component 420 with a threshold period of time. In the preferred implementation, the threshold period of time is set to a fixed period of time. Alternatively, the threshold period of time may be a configurable threshold period of time such that the threshold period of time may be set to any period of time.

When the comparing component 430 determines that the threshold period of time has been exceeded, terminating component 320 generally terminates the partially-completed connection transaction. In one example, terminating component 320 includes a reset component that is structured and arranged to reset a communication port located on the access provider 190. In the instance when the partially-completed connection transaction is based on TCP communications, the communication port on the access provider 190 may be reset when the port remains in existence in excess of the threshold period of time which typically occurs when access provider 190 does not receive a reply (ACK) from access requestor 110. When the communication port is reset, it becomes available for use in response to a new access request (SYN REQ).

Additionally or alternatively, when the comparing component 430 determines that the threshold period of time has been exceeded, terminating component 320 may delay terminating the partially-completed connection transaction to allow the monitoring component 310 to continue monitoring communications with the access provider 190. The terminating component 320 may also block any future access requests from a particular illegitimate access requestor 110 through the use of a header that identifies the IP address of the illegitimate access requestor 110.

Referring to Fig. 5, a process 500 is described for securing an access provider 190, which process 500 may be performed by the systems described above with respect to Figs. 1-4. For instance, the process 500 may be performed by a switch 170, by an access provider 190, or by a combination of the two. The process also may be performed by any other hardware device or software device capable of being programmed to receive, process, and send instructions in the manner described. The process 500 generally includes monitoring communications with an access provider 190 for partially-completed connection transactions (step 510) and terminating the partially-completed connection transactions when the partially-completed connection transactions remain in existence for a period of time that exceeds a threshold period of time (step 520).

In one example, step 510 includes monitoring at least one access provider 190 to detect partially-completed connection transactions by monitoring communications performed to establish a connection involving the access provider 190 based on TCP protocol. In this instance, an access requestor 110 sends an access request (SYN REQ). In response to the access request, the access provider 190 opens a communication port and sends an acknowledgement (SYN ACK) to the access requestor 110. The partially-completed connection transaction may be detected based on a failure of the access requestor 110 to send a reply (ACK) to the access provider 190. For instance, the monitoring step 510 may monitor communication ports to determine whether a connection port that has been opened in response to the access request by the access provider 190 remains in existence beyond the threshold period of time. Additionally or alternatively, step 510 may include monitoring communications with multiple access providers 190 for partially-completed connection transactions.

Step 520 generally includes terminating the partially-completed connection transaction when the partially-completed connection transaction remains in existence for a fixed period of time (e.g., six seconds). Alternatively or additionally, the threshold period of time may be configurable such that it may be set for any period of time.

Referring to Fig. 6, monitoring communications with the access provider 190 for partially-completed connection transactions (step 510 of Fig. 5) may include detecting partially-completed connection transactions (step 610), measuring the period of time that a partially-completed connection transaction remains in existence (step 620), and comparing the period of time with a threshold period of time (step 630).

Detecting 610 may include detecting partially-completed connection transactions that occur when an access requestor 110 initiates a connection transaction and the access requestor 190 subsequently fails to send a reply. In one scenario described above, the detecting 610 may include determining whether an illegitimate access requestor 110 has initiated a connection transaction based on a spoofed return address (e.g., IP address) that differs from the actual return address (e.g., IP address) of the illegitimate access requestor 110. Where the access requestor 110 is a client and the access provider 190 is a host, detecting 610 may include detecting partially-completed connection transactions between at least one client and at least one host. Additionally or alternatively, detecting may include detecting partially-completed connection transactions between at least one client and multiple hosts and/or multiple clients and at least one host. Detecting also may include recognizing other connection transactions or their components (e.g., IP, TCP/IP, UDP, and UDP/IP).

Measuring 620 generally includes measuring the period of time that a partially-completed connection transaction remains in existence. In the example of a partially-completed connection transaction based on TCP communications, the access provider 190 opens a communication port when it receives an access request (SYN REQ) from an access requestor 110. In the instance when the partially-completed connection transaction is based on TCP communications, measuring may measure a period of time that starts when an access request (SYN REQ) is first received. Alternatively, measuring may measure a period of time that starts when the access provider 190 opens a communication port and sends an acknowledgement (SYN ACK) in response to an access request (SYN REQ). Measuring determines the length of the period of time that the communication port remains in existence. Measuring also may include recording the period of time that the communication port remains in existence. Additionally or alternatively, measuring 620 may include measuring the period of time that other types of partially-completed connection transactions (e.g., IP, TCP/IP, UDP, and UDP/IP) remain in existence.

Comparing 630 generally includes comparing the period of time with a threshold period of time. The threshold period of time may be configurable. If the period of time exceeds the threshold period of time, step 520 terminates the partially-completed connection transaction. Terminating 520 may include resetting the communication port on the access provider 190 that was opened in response to an access request initiated by access requestor 110. In the instance when the partially-completed connection transaction is based on TCP

communications, the communication port on the access provider 190 may be reset when the port remains in existence in excess of the threshold period of time which typically occurs when access provider 190 does not receive a reply (ACK) from access requestor 110. When the communication port is reset, it becomes available for use in response to a new access request (SYN REQ). Additionally or alternatively, terminating 520 may delay resetting the communication port to allow monitoring 510 to continue monitoring communications with the access provider 190. Terminating 520 also may block future access requests from an illegitimate requestor 110 through the use of a header that identifies the IP address of the illegitimate access requestor 110.

A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, advantageous results still could be achieved if steps of the disclosed techniques were performed in a different order and/or if components in the disclosed systems were combined in a different manner and/or replaced or supplemented by other components.

In addition, the systems, methods, and techniques described here may be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Apparatus embodying these techniques may include appropriate input and output devices, a computer processor, and a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor. A process embodying these techniques may be performed by a programmable processor executing a program of instructions to perform desired functions by operating on input data and generating appropriate output. The techniques may advantageously be implemented in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program may be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language may be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. Storage devices suitable for tangibly embodying computer program

instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as Erasable Programmable Read-Only Memory (EPROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and Compact Disc Read-Only Memory (CD-ROM disks). Any of the foregoing may be supplemented by, or incorporated in, specially-designed ASICs (application-specific integrated circuits).

Accordingly, other embodiments are within the scope of the following claims.